Combating Data Laundering in LLM Training

Muxing Li[*]
University of Melbourne
muxingl@student.unimelb.edu.au

Zesheng Ye[*]
University of Melbourne
zeshengye.ml@gmail.com

Sharon Li
University of Wisconsin-Madison
sharonli@cs.wisc.edu

Feng Liu[†]
University of Melbourne
fengliu.ml@gmail.com

Abstract

Post-hoc unauthorized-training data detection for *large language models* (LLMs) typically assumes a query-with-originals regime: data copyright holders query a *target* LLM with raw proprietary data and assess whether the model assigns them stronger memorization-based detection signals, e.g., higher confidence or lower loss, than held-out non-training reference texts. We show that this regime becomes brittle under data laundering, where the *target* LLM is trained on semantics-preserving but stylistically or structurally transformed surrogates of proprietary data to obfuscate provenance. Since training-time exposure occurs in the laundered form, memorization signals may no longer appear on the originals, collapsing the candidate-reference signal separation that standard detectors rely on. We counter this threat by studying laundering-aware detection with raw proprietary data, a held-out reference corpus, and query access to the *target* LLM, while the laundering transformation is undisclosed. Since exact recovery of the laundered corpus is infeasible, we infer a detection-useful synthesis process via an auxiliary LLM that maps originals into training-like queries. To make this search tractable, we introduce Synthesis Data Reversion (SDR), which constrains the unbounded space of natural-language transformations through a goal-detail decomposition: a high-level transformation goal, e.g., "lyrical rewriting", and fine-grained details, e.g., "with vivid imagery". SDR identifies the most likely goal and iteratively refines details so synthesized queries elicit stronger target-model detection signals. Evaluated on the MIMIR benchmark against diverse laundering practices and target LLM families (Pythia, Llama2, and Falcon), SDR consistently restores detection signals, offering a practical auditing tool against data laundering. The code is available at https://github.com/MuxingLiUnimelb/SDR.

1 Introduction

Large language models (LLMs) can generate fluent and stylistically diverse text, driving adoption in medicine [22], education [35], and other high-stakes domains. These capabilities require training on large, high-quality corpora [30], whose collection and use are often constrained by privacy and copyright [16]. A central compliance question is therefore whether a deployed LLM was trained on copyrighted or sensitive data without authorization.

Post-hoc *unauthorized training data detection* addresses this question by querying a *target* LLM with proprietary candidate texts and comparing their memorization-based detection signals, such as loss [39] or calibrated confidence [34], against those of a held-out non-training reference corpus,

[*] Equal contribution. [†] Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

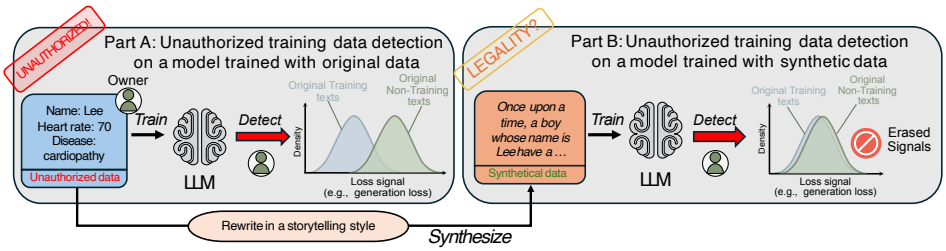


Figure 1: Illustration of how data laundering undermines existing unauthorized training data detectors. When unauthorized data is directly used for training, LLMs tend to memorize the unauthorized training data. Original training samples exhibit lower loss than original non-training samples, as shown in Part A. The log-likelihood distributions of training and non-training samples diverge clearly, enabling identification. However, when trained on laundered synthetic data, as shown in Part B, the distributions of original training samples and original non-training samples no longer diverge, preventing reliable identification.

following Carlini et al. [8]. Since LLMs can memorize training data [17], training samples often receive lower loss or higher confidence than non-training samples. This gap in detection signals enables reliable detection when the target model was trained on the original proprietary texts (Figure 1 (Part A)); mainstream methods perform well in this "query with originals" regime (Table 1, "Orig." columns).

This regime implicitly assumes that, if the target LLM used the rights owner's data, it was exposed to the data in its original textual form. Natural language, however, can be extensively rewritten while preserving its underlying semantics [4, 5]. Such rewrites can be produced through human writing, programmatic paraphrase, back-translation [10, 3], or LLM-based synthesis [32, 21]. If a model provider trains only on these transformed surrogates, the target LLM may memorize the surrogates rather than the originals. As a result, querying with the originals no longer yields a reliable gap in detection signals (Figure 1 (Part B)). Empirically, when a Llama-2 model [29] is trained on Wikipedia articles rewritten into a lyrical style, standard detection methods applied to the originals perform close to random chance (Table 1, "Syn." columns).

We call this evasion mechanism *data laundering*: the model provider transforms proprietary data into semantics-preserving surrogates, trains on the surrogates, and can later claim that the originals were never used. Crucially, data laundering need not compromise training utility, as our experiments show comparable downstream performance to training on original data (see Section 5.2), supporting its practical viability as an evasion strategy. This creates practical risks for copyright and privacy auditing. Data rights owners usually have only black-box API access to the target LLM, and providers rarely disclose preprocessing pipelines, data sources, or training artifacts. The laundering transformation, if it exists, is therefore hidden from the auditor. We ask:

> *How can a data rights owner detect unauthorized data use from black-box model access when the data has been laundered through an unknown transformation?*

Our key idea is to synthesize queries that resemble the laundered data seen during training. If the target model was trained on transformed surrogates, then querying it with similar surrogates should restore the gap in memorization-based detection signals. Since the true transformation is unknown, we use an *auxiliary* LLM as a generator [18] and search for a prompt that maps original texts into training-like rewrites. The challenge is that the space of possible rewriting prompts is too large to search directly.

To make this search tractable, we decompose the unknown transformation into a coarse *goal* and fine-grained *details*. The goal captures the dominant register or structural shift, such as rewriting into lyrics, storytelling, or instructions. The details specify additional constraints, such as imagery, voice, formatting, or rhyme density. This goal-detail decomposition follows the structure of common prompt templates [23]: the goal defines the main rewriting direction, while the details refine the generated surrogates. We use this decomposition only as a search strategy for producing training-like queries, not as a claim that every real laundering pipeline must follow this exact form.[2]

[2] A register is a situational variety of language shaped by purpose, audience, and medium; examples include news, academic prose, instructions, and lyrics [1].

Table 1: Performance of unauthorized training data detection on Llama-2 [29] models fine-tuned with either the original MIMIR-Wikipedia [9] dataset (Orig.) or its laundered version (Syn.) generated by GPT-4o-mini [13] using the prompt "rewrite in a lyrical style, ensuring the imagery is vivid". Evaluation metrics are defined in Section 5.

Methods	AUC		ASR		TPR@5%	
	Orig.	Syn.	Orig.	Syn.	Orig.	Syn.
Loss [37]	1.000	0.539	1.000	0.565	1.000	0.040
Ref [8]	0.971	0.603	0.920	0.610	0.850	0.100
Zlib [7]	1.000	0.521	1.000	0.535	1.000	0.080
Min-K [26]	1.000	0.563	1.000	0.575	1.000	0.040
Recall [34]	0.999	0.558	0.995	0.565	1.000	0.000

Building on this decomposition, we introduce *synthesis data reversion* (SDR), a two-stage method for laundering-aware detection. First, SDR identifies a coarse laundering goal by searching over a bounded set of linguistic registers and selecting the one whose synthesized queries best match the target model's black-box behavior. Second, SDR refines the selected goal by using target-model feedback to infer missing details and update the synthesis prompt. The final prompt produces training-like surrogates for both the proprietary candidate set and the held-out reference set, enabling off-the-shelf detectors, such as Recall [34], to be applied without modification.

We evaluate SDR on MIMIR benchmark [9] across target LLM families including Pythia [6], Llama-2 [29], and Falcon [2], as well as auxiliary LLM choices including DeepSeek [20], GPT-4o-mini [25], and Claude [33]. Across diverse simulated laundering procedures, SDR substantially improves the average performance of standard detection methods, and ablation studies show that both stages contribute to these improvements.

Contributions. This paper makes three contributions. First, we formulate data laundering-aware post-hoc unauthorized training data detection for black-box LLMs. Second, we introduce a goal-detail decomposition that turns the open-ended transformation search into a tractable synthesis problem. Third, we propose SDR, a two-stage method that restores the effectiveness of standard detection methods under data laundering. Together, these results provide a practical auditing approach for data rights holders and highlight data laundering as an emerging risk for LLM governance.

2 Unauthorized Data Detection: Progress and Challenge

Verifying the provenance of data used to train LLMs is a central problem for trustworthy AI [14, 41, 28]. Existing solutions can be broadly divided into proactive defenses and post-hoc detection.

Progress in unauthorized data detection. Proactive defenses aim to prevent or trace misuse before or during training. Data watermarking embeds signals, such as stylistic patterns, into training data so that later model outputs can reveal provenance [19]. Differential privacy [11] provides formal guarantees against memorization by adding calibrated noise during training, but it often reduces model utility and is rarely adopted in LLM training. More importantly, these approaches require action before training and cannot audit already deployed models whose data sources are undisclosed.

Post-hoc detection instead asks whether a deployed model has used specific data after the model has been trained. This setting is especially relevant for LLM auditing, where data rights owners often have only black-box access to the target model. Most post-hoc methods build on membership-inference ideas [27]: training samples tend to induce different model behaviors from unseen samples because overparameterized models can memorize their training data [17]. In LLMs, this difference is commonly measured through intrinsic signals such as loss, likelihood, or calibrated confidence [7, 8, 36, 26, 39, 34]. These methods provide practical tools for detecting unauthorized data use and remain effective in the standard setting where the queried texts match the texts seen during training.

Data laundering and its practicality. Existing post-hoc detectors are designed and evaluated under the "query with originals" regime: the rights holder queries the target LLM with original proprietary texts and compares their signals against those of a non-member reference corpus. This regime breaks down when the target model was trained not on the originals, but on semantics-preserving surrogates. Such surrogates can be produced through paraphrasing, back-translation [4, 3], register or style transfer, or large-scale LLM-based rewriting [32, 38]. We refer to this evasion mechanism

as *data laundering*. Crucially, the practicality of this strategy depends on whether the laundered data remain useful for model training. If laundering degrades downstream task performance, an adversary may have little motivation to adopt it. Our empirical results show that models trained on laundered data retain comparable downstream task performance to those trained on the original data (see Section 5.2).

Challenges posed by data laundering. Under data laundering, the target model may memorize the transformed surrogates rather than the original texts. As a result, original member samples can lose their signal advantage over non-members, causing the candidate-reference separation used by standard detectors to collapse. The challenge is further amplified by the opacity of real-world LLM deployment: data rights owners usually do not know whether laundering occurred, what transformation was used, or what surrogate texts appeared in training.

3 Reverse-engineering the Laundering Transformation

The challenge above suggests a direct but unrealistic solution: recover the exact laundered training texts and query the target model with them. Under black-box access, however, exact recovery is infeasible because the laundering transformation is hidden and the space of semantics-preserving rewrites is too large for brute-force search. We therefore seek a weaker but sufficient goal: a *detection-useful* synthesis process that maps both proprietary candidate texts and held-out reference texts into queries close enough to the target model's laundered training distribution to restore candidate-reference signal separation. We then formalize this objective below.

Problem setup. Let M_t denote the target LLM, trained on an unknown corpus D_{train}, and let M_a denote an auxiliary LLM used by the data rights owner for synthesis. The rights owner has a proprietary candidate corpus D_{pro} and a held-out non-training reference corpus D_{held}. The target training corpus may contain laundered variants of a subset of D_{pro}, but the rights owner has access only to D_{pro}, D_{held}, and black-box queries to M_t. The owner does not observe D_{train}, the laundered corpus, or the provider's preprocessing pipeline.

A synthesis prompt p, executed by M_a, induces a prompt-based transformation $\text{Syn}_p(\cdot)$. Applying this transformation to the candidate and reference corpora yields two synthesized sets, $\text{Syn}_p(D_{\text{pro}})$ and $\text{Syn}_p(D_{\text{held}})$. These sets are then evaluated on M_t using an off-the-shelf unauthorized training data detector.

Objective. Our goal is to find a prompt p that makes the synthesized candidate and reference sets maximally separable under the chosen detector:

$$p^\star = \arg\max_p \text{Perf}\big(M_t, \text{Syn}_p(D_{\text{pro}}), \text{Syn}_p(D_{\text{held}})\big), \tag{1}$$

where $\text{Perf}(\cdot)$ denotes the detector's performance measure, such as AUC, ASR, or TPR at a fixed false-positive rate. A high value of Perf means that the synthesized candidate texts elicit stronger training-data signals than the synthesized reference texts. In this case, p serves as a detection-useful approximation to the hidden laundering transformation: it need not recover the exact training-time surrogates, but it produces queries that restore the signal separation needed by standard detectors. The detailed threat model and objective are provided in Appendix C.

Managing the prompt search space. The objective above still involves searching over natural-language prompts, whose space is effectively unbounded [40]. We therefore restrict the search by decomposing each synthesis prompt into a coarse *goal* and fine-grained *details*. This decomposition follows common prompt-template structures [23] and linguistic theory [1, 24], where a prompt contains a core directive and additional modifiers such as context and constraints. In our setting, the *goal* specifies the dominant transformation direction, such as rewriting the input in a lyrical style, while the *details* specify supplementary requirements that refine the generated surrogate, such as vivid imagery, narrative voice, or formatting constraints.

To make the goal search finite, we ground the goal in a linguistic register taxonomy. Specifically, we search over 23 registers (collected in a set R) that cover major communicative forms [12], using each register as a candidate coarse transformation goal. The remaining problem is then to select the register most compatible with the target model's black-box behavior and refine the details within that register.[3] This leads to the two-stage method in Section 4.

[3]We acknowledge that this taxonomy was not designed for data laundering and thus has limitations, which we discuss in Section 6.

Algorithm 1 Goal identification stage

Require: Proprietary originals D_{pro}, held-out data D_{held}, target model M_t, auxiliary LLM M_a, set of 23 registers R, sample size n and m
Ensure: the register $r^* \in R$ that is closely aligned with the laundering directive.
1: **—Constructing opening templates—**
2: **for all** $r \in R$ **do**
3: Standard-prompt$_r \leftarrow M_a$("Give me a prompt that can transfer text into register r")
4: $O_r \leftarrow \{$ The first sentence of M_a(Standard-Prompt$_r, s$) $\mid s \in$ UniformSample$(D_{\text{pro}}, n)\}$
5: $\text{T}_r \leftarrow M_a$("Extract a common template.", O_r)
6: **end for**
7: **—Scoring via continuation confidence—**
8: **for all** $r \in R$ **do**
9: $S \leftarrow \{$ The first sentence of $s \mid s \in$ UniformSample$(D_{\text{pro}}, m)\}$
10: $X_r \leftarrow \{ M_a$("Rewrite x as T_r") $\mid x \in S \}$
11: **for** $j \leftarrow 1$ **to** m **do**
12: $c_j \leftarrow$ Average next token confidence of $M_t(X_r[j])$
13: **end for**
14: $\text{Conf}(r) \leftarrow \frac{1}{m} \sum_j c_j$
15: **end for**
16: $C \leftarrow$ top-5 registers with largest $\text{Conf}(r)$
17: **—Selecting best register—**
18: **for all** $r \in C$ **do**
19: $\text{Syn}_\text{r} \leftarrow \{ M_a$(Standard-prompt$_r, d) \mid d \in D_{\text{pro}} \cup D_{\text{held}} \}$
20: $\text{Perf}_r \leftarrow$ Unauthorized training data detection on M_t using Syn_r
21: **end for**
22: $r^* \leftarrow \arg\max_{r \in C} \text{Perf}_r$
23: **return** r^*, Standard-prompt$_{r^*}$

4 Synthesis Data Reversion

We propose *synthesis data reversion* (SDR), a two-stage method for searching the prompt p defined in Section 3. Given the goal-detail decomposition, SDR first identifies a coarse rewriting goal and then refines the missing details within that goal. The output is a prompt that synthesizes "training-like" variants of both D_{pro} and D_{held}, so that standard unauthorized training data detectors can be applied to the synthesized corpora. The proposed SDR proceeds in two stages: goal identification and detail refinement, described in Algorithms 1 and 2, respectively.

4.1 Goal Identification Stage

The first stage selects the register r^* that is most compatible with the hidden laundering goal (Algorithm 1). A direct solution would synthesize all samples in $D_{\text{pro}} \cup D_{\text{held}}$ into every register and run the detector for each one. This is expensive because it requires many long-form rewrites. Instead, SDR uses opening sentences as a cheap proxy: if the true laundering process rewrites data into a particular register, then openings written in that register should look familiar to the target model and elicit high-confidence continuations. For example, if the hidden laundering prompt rewrites Wikipedia text into a lyrical style, an opening such as "In the heart of ..." should be more compatible with the target model than an opening written as a legal clause or a recipe. To implement this idea, the first block in Algorithm 1 is designed to construct opening templates.

Constructing opening templates. Lines 2–6 of Algorithm 1 build an opening template for each register $r \in R$. For each r, SDR first asks the auxiliary LLM M_a to produce a *Standard-prompt* that transfers text into that register (line 3). It then applies this standard prompt to n sampled proprietary texts and keeps only the first sentence of each rewrite (line 4). Finally, M_a abstracts these first sentences into a common opening template T_r (line 5). For instance, for the lyrical register, the sampled rewrites may begin with vivid, image-heavy openings; M_a may summarize them into a template such as "In the heart of [abstract domain], a tale unfolds, where ...". This template captures how the register typically begins without rewriting the full corpus.

Scoring registers by continuation confidence. Lines 8–16 use the target model M_t to score how compatible each register is with the training-time distribution. For each register r, SDR samples m proprietary texts and extracts their original first sentences (line 9). It then rewrites these first sentences according to the template T_r, producing short register-conditioned queries X_r (line 10).

Algorithm 2 Detail refinement stage

Require: register r^* and Standard-prompt$_{r^*}$ got from Algorithm 1 , proprietary data D_{pro}, held-out data D_{held}, target model M_t, auxiliary LLM M_a, iteration budget K, sample size l
Ensure: Reversed prompt
1: $p =$ Standard-prompt$_{r^*}$
2: **function** CONDITIONINFERENCE($D_{\text{pro}}, p, M_t, M_a$)
3: **for all** $s \in$ UniformSample(D_{pro}, l) **do**
4: $\hat{s} \leftarrow M_a(p, s)$
5: $\tilde{s} \leftarrow M_t$(the first sentence of $\hat{s}$)
6: H.APPEND(M_a("Editing p enables the transformation of $\hat{s}$ into $\tilde{s}$"))
7: **end for**
8: **return** M_a("Extract a common prompt.", H)
9: **end function**
10: **function** EVALUATE($D_{pro}, D_{\text{held}}, p, M_t, M_a$)
11: $\text{Syn}_{\text{p}} \leftarrow \{ M_a(p, x) \mid x \in D_{\text{pro}} \cup \mathcal{D}_{\text{held}} \}$
12: $\text{Perf}_p \leftarrow$ Unauthorized training data detection on M_t using Syn_{p}
13: **return** Perf_p
14: **end function**
15: **for** $k \leftarrow 0$ **to** $K - 1$ **do**
16: $p' \leftarrow$ CONDITIONINFERENCE($D_{\text{pro}}, p, M_t, M_a$)
17: $\text{score}_{\text{new}} \leftarrow$ EVALUATE($D_{\text{pro}}, D_{\text{held}}, p', M_t, M_a$), $\text{score}_{\text{old}} \leftarrow$ EVALUATE($D_{\text{pro}}, D_{\text{held}}, p, M_t, M_a$)
18: **if** $\text{score}_{\text{new}} > \text{score}_{\text{old}}$ **then**
19: $p \leftarrow p'$
20: **end if**
21: **end for**
22: **return** p

Each query $X_r[j]$ is fed into M_t where $j = 1, \ldots, m$, and SDR measures how confidently the target model continues it (lines 11–14). Specifically, let the continuation length for $X_r[j]$ be L_j, and let $X^{(j)}_{r,<i}$ denote the rewritten opening sentence together with the first $i-1$ generated tokens. We compute

$$\text{Conf}(r) = \frac{1}{m} \sum_{j=1}^{m} \left(\frac{1}{L_j} \sum_{i=1}^{L_j} \max_{w \in V} P_{M_t}\left(w \mid X^{(j)}_{r,<i} \right) \right), \tag{2}$$

where V is the vocabulary, and $P_{M_t}(w \mid X^{(j)}_{r,<i})$ denotes the probability assigned by the target model M_t to token w as the next token given the current prefix $X^{(j)}_{r,<i}$. The intuition is that a register closer to the laundering goal should make M_t more confident when continuing the opening. SDR keeps the top-5 registers with the largest $\text{Conf}(r)$ as a small candidate set C (line 16).

Selecting the best register. Lines 18–22 choose the final goal from the shortlisted registers in C. For each $r \in C$, SDR uses the corresponding *Standard-prompt* (obtained in line 3) to synthesize both the proprietary candidate corpus and the held-out reference corpus (line 19). It then runs the chosen unauthorized training data detector on M_t using these synthesized queries and records the resulting performance Perf_r (line 20). The selected register r^* is the one that maximizes this detector performance (line 22), and SDR returns both r^* and its *Standard-prompt* (line 23). Thus, stage 1 turns the open-ended goal search into a finite register search and provides the initial prompt for stage 2.

4.2 Detail Refinement Stage

The selected register gives only a coarse transformation direction. For example, knowing that the laundering goal is "lyrical rewriting" does not specify whether the rewrite should use vivid imagery, first-person narration, rhyme, short lines, or a particular narrative voice. Such missing details can matter because the target model was trained on a specific surrogate distribution, not merely on any text from the same broad register. The second stage therefore refines the prompt returned by Algorithm 1 using feedback from M_t (Algorithm 2).

Initializing the prompt. Line 1 of Algorithm 2 initializes the current prompt p as Standard-prompt$_{r^*}$, the standard prompt associated with the selected register. This prompt already rewrites in the coarse target direction, but it may miss the fine-grained details to produce detection-useful queries.

Inferring candidate refinements from target continuations. Lines 2–9 define CONDITIONINFERENCE. For each sampled proprietary text s, SDR first synthesizes a surrogate $\hat{s} = M_a(p, s)$ using the current prompt p (line 4). It then feeds the first sentence of $\hat{s}$ into the target model M_t and obtains a continuation $\tilde{s}$ (line 5). The first sentence serves as a trigger: if the current prompt is close to the training-time style, the target model's continuation may reveal stylistic or structural properties of the laundered data. SDR then asks M_a to describe how p should be edited so that $\hat{s}$ better matches $\tilde{s}$ (line 6). For example, if $\hat{s}$ is a generic lyrical rewrite but $\tilde{s}$ continues with denser imagery and stronger rhythm, M_a may propose adding details such as "use vivid imagery" or "make the rhythm flow naturally." After collecting such suggestions over l samples, SDR asks M_a to distill them into a single revised prompt p' (line 8).

Evaluating a refinement. Lines 10–14 define EVALUATE. Given a prompt p, SDR synthesizes both D_{pro} and D_{held} with M_a (line 11), runs the unauthorized training data detector on M_t using the synthesized corpora (line 12), and returns the resulting score Perf_p (line 13). This evaluation step is essential: a refinement is useful only if it improves candidate-reference separation, not merely if it sounds closer to the target model's continuation.

Iterative prompt refinement. Lines 15–21 repeat the refinement process for at most K iterations. At each iteration, SDR proposes a new prompt p' through CONDITIONINFERENCE (line 16), evaluates both the proposed prompt and the current prompt (line 17), and accepts the proposal only when $\text{score}_{\text{new}} > \text{score}_{\text{old}}$ (lines 18–19). This acceptance rule keeps the search aligned with the objective in Section 3: the prompt is refined only when it improves detection performance on synthesized candidate and reference sets. The final returned prompt p (line 22) is therefore a detection-useful synthesis prompt that combines a coarse register goal with refined details.

> **Remark.** SDR does not aim to recover the exact laundering prompt used by the model provider. Instead, it seeks a *detection-useful* prompt that restores candidate-reference signal separation under standard detectors. The recovered prompt may therefore differ from the true one, so its success should be evaluated by detection performance rather than exact prompt matching. The next section empirically tests this criterion across different laundering prompts, datasets, target models, and auxiliary LLMs, including settings where the auditor's auxiliary LLM M_a differs from the model used for laundering.

5 Experiments and Results

We first describe the datasets, target models, baselines, metrics, and laundering protocol.

5.1 Experimental Setup

Datasets and target models. We evaluate SDR on the MIMIR benchmark [9] (see Appendix F for more details), which is widely used for unauthorized training data detection. We use three subsets from MIMIR: Wikipedia, C4, and HackerNews, corresponding to encyclopedia articles, web text, and news-style discussions, respectively. Unless otherwise specified, the main experiments use MIMIR-Wikipedia and a Pythia-6.9B [6] target model. To test robustness, we further evaluate different target architectures, including Pythia-6.9B [6], Falcon-7B [2], and Llama-2-7B [29].

Baselines and metrics. We evaluate SDR as a plug-in procedure for five off-the-shelf unauthorized training data detectors: Loss [37], Ref [8], Zlib [7], Min-K [26], and Recall [34]. Following prior work [8], we report Area Under the Curve (AUC), Attack Success Rate (ASR), and True Positive Rate at 5% False Positive Rate (TPR@5%). AUC measures overall separability, ASR measures single-threshold detection accuracy, and TPR@5% captures performance in a low-false-positive auditing regime. Detailed definitions are provided in Appendix G.1.

Laundering prompts and evaluation protocol. We simulate data laundering by rewriting original MIMIR samples into synthetic surrogates before target-model fine-tuning. We consider two prompt families. *Inside-register prompts* rewrite data into one of the 23 registers used by SDR, while *outside-register prompts* are generated to fall outside this predefined taxonomy. The full prompt lists are provided in Appendix H. For each laundering prompt, the model provider synthesizes the original data, fine-tunes the target model on a subset of the synthesized data, and keeps the remaining synthesized samples as held-out references. The auditor only has access to the original candidate data, the held-out reference data, an auxiliary LLM, and black-box queries to the target model. SDR then infers a prompt,

Table 2: The average performance of each unauthorized training data detection method across data synthesized from **different inside and outside laundering processes**. The experiment is conducted on Pythia-6.9B [6], fine-tuned on Wikipedia synthesis. The results for each prompt are provided in Appendix I. We report the additional TPR@1% results in Appendix P.

Method	Inside registers			Outside registers		
	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%
Recall	65.6%	64.8%	13.0%	61.8%	63.0%	8.2%
Recall+SDR	**79.5%**	**75.2%**	**35.5%**	**74.0%**	**70.7%**	**22.6%**
Loss	64.4%	63.6%	11.0%	62.1%	62.3%	8.8%
Loss+SDR	**80.7%**	**76.3%**	**40.0%**	**75.4%**	**71.8%**	**22.8%**
Ref	69.0%	66.8%	14.1%	66.8%	64.8%	12.8%
Ref+SDR	**77.0%**	**72.8%**	**34.8%**	**69.4%**	**66.4%**	**20.0%**
Zlib	64.9%	64.3%	16.4%	63.0%	63.2%	13.9%
Zlib+SDR	**72.1%**	**69.6%**	**22.8%**	**68.5%**	**66.8%**	**18.0%**
Min-K	64.8%	63.5%	11.0%	64.2%	62.7%	10.4%
Min-K+SDR	**78.8%**	**74.7%**	**34.3%**	**75.2%**	**71.6%**	**24.5%**

rewrites both candidate and reference sets, and runs the off-the-shelf detector on the synthesized queries. Further implementation details are provided in Appendix G.

5.2 Downstream Utility of Laundered Data

The practicality of data laundering depends on whether transformed training data preserve their utility for downstream tasks. To investigate this, we evaluate whether models fine-tuned on laundered data retain performance comparable to those fine-tuned on the original data.

We conduct experiments on two base models, Llama-2-7B and Pythia-6.9B, using the MIMIR-Wikipedia and MIMIR-Arxiv datasets. For each dataset, we employ GPT-4o-mini [13] to generate cloze-style questions from the original texts. We then reformulate these questions to assess the factual knowledge in the texts. We compare three settings: (1) the base model without fine-tuning (Base), (2) the model fine-tuned on the original data (Orig.), and (3) models fine-tuned on laundered data generated using each of the 23 inside-register transformation prompts listed in Table 12 (Laund.). All models are evaluated on the same question set using contains-answer accuracy, which measures whether the generated response contains the correct answer. We report the average accuracy across the 23 laundered-data fine-tuned models.

Table 3: Downstream cloze-style questions accuracy (%) of base models and models fine-tuned on original or laundered data. Laundered results are averaged over 23 inside-prompts. Detailed results for each prompt are provided in Appendix D.

Model	Dataset	Base	Orig.	Laund.
Llama-2	Wikipedia	6.8	13.7	13.2
	Arxiv	6.4	10.2	10.12
Pythia	Wikipedia	10.0	13.2	12.8
	Arxiv	8.3	9.8	9.4

As shown in Table 3, fine-tuning on laundered data consistently outperforms the base models and achieves performance comparable to fine-tuning on original data across all four settings, with accuracy differences of only 0.08–0.5 percentage points. Notably, some laundering prompts even yield better downstream performance than the original data (see Appendix D for detailed results). These findings suggest that data laundering can preserve downstream task utility, supporting its practicality for obscuring training data provenance.

5.3 Main Result: SDR Restores Detection under Diverse Laundering Prompts

Table 2 reports the main results on MIMIR-Wikipedia with a Pythia-6.9B target model under both inside-register and outside-register laundering prompts. Across both prompt families, SDR consistently improves the average performance of standard detectors, particularly in terms of TPR@5%. These results support that exact prompt recovery might be unnecessary for auditing. Even when the recovered prompt differs from the true laundering prompt, the synthesized queries can restore candidate-reference separation and make existing detectors effective again. Detailed per-prompt

Table 4: Comparison of average performance of unauthorized training data detection across **different datasets** trained with synthesis using outside register prompts.

Method	Wikipedia			Hackernews			C4		
	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%
Recall	61.7%	61.4%	5.6%	53.9%	56.2%	9.9%	52.2%	55.0%	5.0%
Recall+SDR	**73.4%**	**73.3%**	**23.2%**	**61.6%**	**60.3%**	**10.7%**	**65.2%**	**63.0%**	**12.0%**
Loss	62.7%	62.6%	9.2%	54.2%	56.0%	6.3%	58.4%	59.1%	7.2%
Loss+SDR	**75.5%**	**75.5%**	**22.9%**	**62.7%**	**59.6%**	**8.7%**	**67.3%**	**64.2%**	**13.0%**
Min-K	64.2%	62.5%	10.5%	53.8%	55.9%	5.7%	57.6%	59.2%	6.2%
Min-K+SDR	**73.6%**	**73.5%**	**22.7%**	**61.7%**	**60.8%**	**8.3%**	**66.8%**	**64.7%**	**11.8%**

Table 5: Comparison of average performance of unauthorized training data detection across three **different model architectures** fine-tuned with synthesis generated by outside register prompts.

Method	Pythia-6.9B			Falcon-7B			Llama-2-7B		
	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%
Recall	61.7%	61.4%	5.6%	62.1%	61.8%	8.2%	61.6%	61.6%	8.2%
Recall+SDR	**73.4%**	**73.3%**	**23.2%**	**72.4%**	**69.1%**	**23.0%**	**67.5%**	**66.2%**	**12.5%**
Loss	62.7%	62.6%	9.2%	64.4%	64.4%	11.5%	64.5%	64.4%	11.1%
Loss+SDR	**75.5%**	**75.5%**	**22.9%**	**71.2%**	**68.0%**	**20.2%**	**73.6%**	**70.4%**	**26.9%**
Min-K	64.2%	62.5%	10.5%	63.3%	62.6%	10.5%	62.5%	62.5%	13.9%
Min-K+SDR	**73.6%**	**73.5%**	**22.7%**	**70.2%**	**68.0%**	**20.0%**	**70.9%**	**68.2%**	**22.9%**

Table 6: Comparison of the average performance of unauthorized training data detection with SDR using **different auxiliary LLMs**.

Method	GPT-4o-mini			Claude			DeepSeek		
	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%	AUC	ASR	TPR@5%
Recall	64.5%	63.5%	9.9%	66.8%	65.9%	9.7%	65.5%	64.8%	7.4%
Recall+SDR	**79.7%**	**75.0%**	**31.6%**	**81.6%**	**77.7%**	**31.0%**	**79.8%**	**75.5%**	**31.2%**
Loss	63.4%	62.2%	14.3%	67.3%	65.3%	12.1%	65.9%	64.7%	12.2%
Loss+SDR	**80.3%**	**75.6%**	**32.6%**	**83.4%**	**77.8%**	**41.9%**	**79.0%**	**74.7%**	**32.8%**
Min-K	63.7%	62.3%	11.0%	68.1%	65.5%	14.5%	67.0%	64.6%	14.2%
Min-K+SDR	**72.0%**	**75.1%**	**32.8%**	**81.2%**	**75.3%**	**34.7%**	**77.8%**	**73.1%**	**30.3%**

results, including the original and SDR-recovered prompts, are provided in Appendix I. We further test mixed-register laundering prompts in Appendix L, where SDR also improves detection performance.

5.4 Transferability Results: Robustness under Different Scenarios

Robustness across datasets and target models. We next evaluate whether SDR generalizes beyond the default Wikipedia/Pythia setting. Table 4 reports results on three MIMIR subsets under outside-register laundering prompts. SDR also improves detection across all datasets, indicating that SDR is not tied to a single data distribution. Table 5 evaluates robustness across target model architectures. We fine-tune Pythia-6.9B, Falcon-7B, and Llama-2-7B on laundered MIMIR-Wikipedia data generated from outside-register prompts. SDR improves all three detector families across all model architectures, indicating that SDR is not specific to one target architecture.

Robustness to auxiliary-model mismatch. A realistic auditor should not know which model was used to generate the laundered training data. We therefore test whether SDR remains effective when the auxiliary LLM used by the auditor differs from the laundering model. In Table 6, the model provider launders data with GPT-4o-mini, while SDR uses GPT-4o-mini, Claude-haiku-3-5, or DeepSeek-v3 as the auxiliary model. The gains remain considerable, indicating that SDR does not require the auditor's auxiliary model to match the model provider's laundering model. We also evaluate stronger transferability settings in Appendix K. In particular, we test third-party laundering by DeepSeek-v3 and human rewriting using the Polite dataset [31]. SDR remains generally beneficial in these settings, showing that it can produce detection-useful transformations even when the laundering pipeline differs from the auditor's auxiliary model.

5.5 Ablation Study: Both Goal Identification and Detail Refinement Matter

We conduct ablation studies to evaluate the contribution of the two stages in SDR. Figure 2 compares the full method with two variants: removing goal identification (w/o stage 1) and removing detail refinement (w/o stage 2). Both stages contribute substantially to detection performance. Figure 2 also validates the two-stage design: goal identification narrows the search to a plausible register-level direction (with large performance drop after removing stage 1), while detail refinement adjusts the prompt toward a more detection-useful synthesis process. Hyperparameter sensitivity analysis for the iteration budget K, sample sizes l, m, and n are in Appendix M. We further compare SDR with directly optimizing Equation 1 under the same computational budget. Direct prompt search yields limited improvements, whereas SDR consistently achieves substantially higher detection performance, highlighting the importance of its structured two-stage search (see Appendix N).

5.6 Negative Control: SDR Does Not Fabricate Evidence without Laundering

Because SDR searches for prompts that maximize detector performance, it is important to verify that it does not create false evidence when the proprietary data was not used for training. We therefore conduct a negative-control experiment where neither D_{pro} nor any laundered variant of it appears in the target model's training data. In this setting, SDR should fail to find a prompt that improves detection (i.e., AUC and ASR should be near 0.5), which is verified by results in Appendix J.

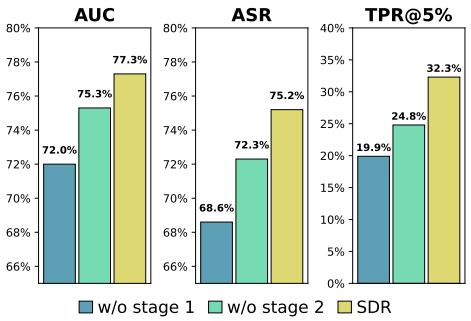


Figure 2: Ablation study on the effectiveness of each stage in SDR. Results are reported as the average performance of unauthorized training data detectors across different inside prompts. Removing the goal identification stage (w/o stage 1) or the detailed prompt condition inference stage (w/o stage 2) leads to noticeable degradation, while the full SDR consistently achieves the best performance.

5.7 Additional Stress Tests

We include several additional evaluations in the appendix. First, Appendix Q evaluates partial laundering, where only a subset of D_{pro} is laundered and used for training; SDR still improves all evaluated detectors. Second, Appendix O compares SDR with an adapted reverse prompt engineering baseline [15]; SDR achieves stronger detection gains, highlighting the difference between recovering inference-time prompts and approximating laundering transformations applied before training.

6 Limitations

A key limitation of our current approach is that the register taxonomy it relies on is coarse-grained to locate goals accurately. Although the existing taxonomy of 23 sub-registers offers broad coverage of textual styles, it was not designed for classifying the laundering goal. Consequently, there are cases where none of the 23 registers can capture the intent of a synthesized prompt, leading to reduced accuracy in goal identification and, in turn, lower quality in restored prompts. Overcoming this limitation calls for future research on developing more fine-grained taxonomies tailored to synthesized data, thereby enabling more accurate and robust prompt reversal in practical scenarios.

7 Conclusion

This paper identifies data laundering as a critical blind spot in post-hoc unauthorized training data detection: when proprietary data is transformed into semantics-preserving surrogates before training, querying the target model with the originals may no longer reveal memorization-based signals. We propose SDR, a two-stage method that searches for a detection-useful synthesis prompt rather than recovering the exact laundering transformation. SDR first identifies a coarse register-level goal and then refines fine-grained details using target-model feedback, producing training-like queries that allow off-the-shelf detectors to be reused. Experiments across diverse scenarios demonstrate that SDR substantially improves the average detection performance of standard methods. These results suggest that detection-useful synthesis is a practical direction for auditing unauthorized data use under data laundering.

8 Acknowledgements and Disclosures

Funding statement. Muxing Li is supported by the ARC with grant number DP230101540. Feng Liu is supported by the ARC with grant numbers DP230101540, DE240101089 and DP260103996.

Competing-interests disclosure. The authors declare no competing interests.

Acknowledgements. This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

References

[1] Asif Agha. Registers of language. *A companion to linguistic anthropology*, 2004.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Etienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The Falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[3] Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *ACL*, 2005.

[4] Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. In *ACL*, 2001.

[5] Rahul Bhagat and Eduard Hovy. What is a paraphrase? *CL*, 2013.

[6] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

[7] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security*, 2021.

[8] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *SP*, 2022.

[9] Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*, 2023.

[10] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP*, 2005.

[11] Cynthia Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[12] Erik Henriksson, Amanda Myntti, Saara Hellström, Anni Eskelinen, Selcen Erten-Johansson, and Veronika Laippala. Automatic register identification for the open web using multilingual deep learning. *arXiv preprint arXiv:2406.19892*, 2024.

[13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[14] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 2023.

[15] Hanqing Li and Diego Klabjan. Reverse prompt engineering. *arXiv preprint arXiv:2411.06729*, 2024.

[16] Haoran Li, Yulin Chen, Jinglong Luo, Jiecong Wang, Hao Peng, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, Zenglin Xu, et al. Privacy in large language models: Attacks, defenses and future directions. *arXiv preprint arXiv:2310.10383*, 2023.

[17] Muxing Li, Zesheng Ye, Yixuan Li, Andy Song, Guangquan Zhang, and Feng Liu. Membership inference attack should move on to distributional statistics for distilled generative models. *arXiv preprint arXiv:2502.02970*, 2025.

[18] Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, et al. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*, 2024.

[19] Yuqing Liang, Jiancheng Xiao, Wensheng Gan, and Philip S Yu. Watermarking techniques for large language models: A survey. *arXiv preprint arXiv:2409.00089*, 2024.

[20] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[21] Qin Liu, Fei Wang, Nan Xu, Tianyi Yan, Tao Meng, and Muhao Chen. Monotonic paraphrasing improves generalization of language model prompting. In *EMNLP Findings*, 2024.

[22] Xiaohong Liu, Hao Liu, Guoxing Yang, Zeyu Jiang, Shuguang Cui, Zhaoze Zhang, Huan Wang, Liyuan Tao, Yongchang Sun, Zhu Song, et al. A generalist medical language model for disease diagnosis assistance. *Nature medicine*, 2025.

[23] Yuetian Mao, Junjie He, and Chunyang Chen. From prompts to templates: A systematic prompt template analysis for real-world llmapps. In *FSE*, 2025.

[24] Amanda Myntti, Erik Henriksson, Veronika Laippala, and Sampo Pyysalo. Register always matters: Analysis of llm pretraining data through the lens of language variation. *arXiv preprint arXiv:2504.01542*, 2025.

[25] Konstantinos I Roumeliotis and Nikolaos D Tselikas. Chatgpt and open-ai models: A preliminary review. *FI*, 2023.

[26] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

[27] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *SP*, 2017.

[28] Xunye Tian, Zhijian Zhou, Liuhua Peng, Jared Collette, Dino Sejdinovic, Yue Yang, and Feng Liu. Tribet: Relative e-values for online machine-generated text detection. In *NeurIPS*, 2026.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[30] Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, et al. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025.

[31] Xun Wang, Tao Ge, Allen Mao, Yuki Li, Furu Wei, and Si-Qing Chen. Pay attention to your tone: Introducing a new dataset for polite language rewrite. *arXiv preprint arXiv:2212.10190*, 2022.

[32] Sam Witteveen and Martin Andrews. Paraphrasing with large language models. In *ACL Workshop on Neural Generation and Translation*, 2019.

[33] Sean Wu, Michael Koo, Lesley Blum, Andy Black, Liyo Kao, Fabien Scalzo, and Ira Kurtz. A comparative study of open-source large language models, gpt-4 and claude 2: Multiple-choice test taking in nephrology. *arXiv preprint arXiv:2308.04709*, 2023.

[34] Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. Recall: Membership inference via relative conditional log-likelihoods. *arXiv preprint arXiv:2406.15968*, 2024.

[35] Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, 2024.

[36] Wentao Ye, Jiaqi Hu, Liyao Li, Haobo Wang, Gang Chen, and Junbo Zhao. Data contamination calibration for black-box llms. *arXiv preprint arXiv:2405.11930*, 2024.

[37] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *CSF*, 2018.

[38] Brook Zeleke, Amish Soni, and Lydia Manikonda. Human or genai? characterizing the linguistic differences between human-written and llm-generated text. In *ACM Web Science Conference*, 2025.

[39] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*, 2024.

[40] Xiang Zhang, Juntai Cao, Jiaqi Wei, Chenyu You, and Dujian Ding. Why prompt design matters and works: A complexity analysis of prompt search space in llms. *arXiv preprint arXiv:2503.10084*, 2025.

[41] Zhexin Zhang, Yuhao Sun, Junxiao Yang, Shiyao Cui, Hongning Wang, Minlie Huang, et al. Be careful when fine-tuning on open-source llms: Your fine-tuning data could be secretly stolen! In *ICLR*, 2026.

A Mathematical Notation

We summarize the key mathematical symbols used throughout the paper.

Symbol	Description
M_t	Target LLM trained on D_{train}.
M_a	Auxiliary LLM used only for synthesis.
D_{pro}	Proprietary dataset (candidate corpus).
D'_{pro}	Unknown in-training subset of D_{pro}.
D_{train}	Model provider's full training dataset.
D_{lau}	Laundered version of D'_{pro}.
D_{held}	Held-out non-training reference dataset.
p	Current reverse-synthesis prompt refined during iterations.
$\mathcal{R}$	Set of 23 established registers.
r^*	Register selected as most closely aligned with the laundering goal.
n, m, l	Sample sizes used in constructing templates, scoring, and inference, respectively.
K	Maximum number of iterations in the detailed prompt condition inference stage.
Standard-prompt$_r$	A canonical prompt that synthesizes text into register r.
T	True (unknown) laundering transformation
T_r	Opening template extracted for register r.
s	Original sample.
$\hat{s}$	Synthetic data generated by M_a from an original sample s under prompt p.
$\tilde{s}$	Continuation produced by the target model M_t when prompted with $\hat{s}$.
$\text{Conf}(r)$	Average next-token confidence of M_t under register r.
$\mathcal{C}$	Candidate set of top-k registers with highest confidence scores.
Syn_r	Dataset synthesized into register r by M_a.
Perf_r	Unauthorized training data detection performance of Syn_r on M_t.
Syn_p	Surrogate set induced by prompt p.
Perf_p	Detector performance under prompt p.

B Register Taxonomy

The register taxonomy proposed by Henriksson et al. [12] defines 23 sub-registers across nine major categories, ranging from narrative forms (e.g., news and sports reports, blogs) and informational texts (e.g., encyclopedia entries, research articles, legal documents) to opinion pieces, persuasive descriptions, interactive discussions (e.g., FAQs, interviews), instructional texts (e.g., recipes), spoken and lyrical registers, and machine-translated content (details are shown in Table 7). This taxonomy provides a broad linguistically motivated coverage of common open-domain registers. In our work, it is used as a systematic and scalable framework that reduces unbounded stylistic variation into a bounded set of functional categories.

Table 7: Register categories and abbreviations.

Name	Abbr.	Name	Abbr.
Lyrical	ly	Encyclopedia article	en
Spoken	sp	Research article	ra
Interview	it	Description of a thing or person	dtp
Interactive discussion	id	FAQ	fi
Narrative	na	Legal terms & conditions	lt
News report	ne	Opinion	op
Sports report	sr	Review	rv
Narrative blog	nb	Opinion blog	ob
How-to or instructions	hi	Denominational religious blog or sermon	rs
Recipe	re	Description with intent to sell	ds
Informational persuasion	ip	Informational description	in
News & opinion blog or editorial	ed		

Table 8: Notation list for the threat model and if they are visible to rights holder.

Symbol	Description & Role	Visible to Rights Holder?
M_t	Target LLM trained on D_{train}	No
M_a	Auxiliary LLM used only for synthesis	Yes
T	True (unknown) laundering transformation	No
p	Reverse-synthesis prompt inferred by SDR	Yes
D_{pro}	Proprietary dataset (candidate corpus)	Yes
D'_{pro}	Unknown in-training subset of D_{pro}	No
D_{train}	Model provider's full training dataset	No
D_{lau}	Laundered version of D'_{pro}	No
D_{held}	Held-out non-training reference dataset	Yes
s	Original sample	Yes
$\hat{s}$	Synthetic surrogate rewrite	Yes
Syn_p	Surrogate set induced by prompt p	Yes
Perf_p	Detector performance under prompt p	Yes

C The Objective and Threat Model

Threat model. Our setting follows a black-box auditing formulation used in post-hoc unauthorized data detection [34, 8] where: 1. The model provider trains the target LLM M_t on an unknown training set D_{train} that may include laundered variants of some samples in a proprietary dataset D_{pro}. 2. The data rights holder owns a proprietary dataset D_{pro} and has a held-out reference set D_{held} guaranteed not to appear in D_{train}. The rights holder only has access to (i) D_{pro}, (ii) D_{held}, and (iii) black-box query access to M_t (e.g., API). It does **not** have access to D_{train}, nor to the (potentially unknown) laundering pipeline. 3. An auxiliary LLM M_a is any model that the rights holder can query to synthesize surrogate samples. Then, we aim to help rights holders detect whether M_t used unauthorized data or laundered variants of unauthorized data for its training. We additionally summarize all symbols and their roles in Table 8.

What SDR optimizes. SDR seeks to "optimize" a prompt p so that samples in D_{pro} are mapped closer to the true laundered variants used in training M_t, thereby restoring the effectiveness of standard unauthorized training-data detectors. Firstly, We use M_a to synthesize $\text{Syn}_p(D_{\text{pro}})$ and $\text{Syn}_p(D_{\text{held}})$. Then we run an off-the-shelf detector (Recall, Min-K, etc.) on M_t using these two sets, exactly as in standard post-hoc unauthorized data detection, but using the synthesized variants. Finally, SDR takes the detector's output scalar score (e.g., Recall's two-sample score) as Perf_p, and SDR seeks to optimize a prompt p that maximizes Perf_p.

Detailed definition of Perf_p. Given a fixed detector (e.g., Recall, Min-K), a prompt p induces synthetic surrogates $\text{Syn}_p(D_{\text{pro}})$ and $\text{Syn}_p(D_{\text{held}})$ via M_a. We then run the detector on M_t precisely as in prior work, but using these surrogates as candidate vs. reference sets, and record a scalar performance measure Perf_{p}. Algorithms 1 and 2 describe our pipeline that searches over the prompt space to maximize this scalar. Perf_{p} distinguishes $\text{Syn}_p(D_{\text{pro}})$ from $\text{Syn}_p(D_{\text{held}})$, implying that M_t was indeed trained on a laundered version of D_{pro}, and p can recover the unknown laundering process used by the model provider. Otherwise, it implies that the current prompt p cannot recover the laundered training data, and we need to continue searching for more plausible ones. Eventually, if no such performance-improving prompt can be found, we consider that no laundering of D_{pro} was used in training M_t.

D Detailed Downstream Utility Results

Table 9 reports the cloze-style question-answering performance of Llama-2-7B fine-tuned on MIMIR-Wikipedia data rewritten using each of the 23 inside-register laundering prompts, following the evaluation protocol in Section 5.2. All 23 models trained on laundered data outperform the base model, with contains-answer accuracy ranging from 8.9% to 16.8%, compared with 6.8% for the base model. Moreover, 11 of the 23 prompts yield higher accuracy than fine-tuning on the original data (13.7%), and one matches it. These results show that data laundering does not necessarily reduce downstream utility relative to fine-tuning on original data, although its effect varies across laundering prompts.

Table 9: Cloze-style question-answering performance of Llama-2-7B fine-tuned on original or laundered MIMIR-Wikipedia data. Prompt descriptions are abbreviated; IDs correspond to the full prompts in Table 12. Bold values exceed the original-data fine-tuning baseline.

Prompt	Fine-tuning setting / laundering style	Contains-answer accuracy (%)
–	Base model (no fine-tuning)	6.8
–	Fine-tuned on original data	13.7
1	Lyrical style	10.3
2	Spoken style	**14.7**
3	Interview	**14.2**
4	Interactive discussion	11.6
5	Storytelling narrative	**14.2**
6	News report	**15.3**
7	Sports report	9.5
8	Narrative blog post	**14.6**
9	Step-by-step instructional guide	**16.8**
10	Recipe	**14.2**
11	Factual persuasion	11.1
12	Sales description	12.4
13	Editorial	11.6
14	Informational description	**15.8**
15	Encyclopedia entry	13.2
16	Academic research article	13.7
17	Descriptive profile	**14.7**
18	Frequently Asked Questions (FAQ)	**14.2**
19	Legal terms and conditions	**16.3**
20	Personal opinion piece	11.6
21	Review	12.6
22	Opinion blog post	8.9
23	Denominational religious sermon	12.6

E Opening Template

Table 10 presents the representative opening templates T_r that we derived for each register. These templates were obtained by synthesizing a small subset of samples into the corresponding register r and then prompting a large language model to extract a generalized first-sentence structure. As shown in the table, each register exhibits distinct stylistic cues in its openings: for example, lyrical texts often begin with abstract imagery, interviews with a direct address from the interviewer, and storytelling narratives with a scene-setting phrase such as “Once upon a time.” Such templates capture the prototypical entry points of different registers, which serve as useful anchors for aligning synthesized outputs with their intended discourse forms.

F An Introduction to the MIMIR Dataset

The MIMIR benchmark is specifically constructed for evaluating unauthorized training data detection. For each model to be evaluated (Pythia, Falcon, Llama-2), it defines "seen" splits (i.e., ground-truth

Table 10: Registers and their corresponding opening templates.

Register	Template (first-sentence / opening)
Lyrical	In the heart of [abstract domain], a tale unfolds, where [abstract concept], [abstract detail], [abstract entity], [abstract action].
Spoken style	So, let's talk about [TOPIC].
Interview	**Interviewer:** Thank you for joining us, [Person/Expert Title]. Can you tell us about [Subject/Topic]?
Interactive discussion	**[Participant 1]:** So, have you guys heard about [Topic/Subject]? I recently came across some interesting information about it.
Storytelling narrative	Once upon a time, in a [adjective] [type of place] called [place name], there lived a [adjective] [type of character] named [character name].
News report	**[Event/Topic]: [Description/Significance] [Location/Context]** – [Details about the subject, including noteworthy contributions, roles, or milestones].
Sports report	In a thrilling [event/display/action], [subject/actor] has [verb] [description/impact] in [field/area/genre].
Narrative blog post	In the context of [broad category or field], [subject or specific work] has made a significant impact, often leading to [general observation or effect].
Step-by-step guide (How-to)	**Step-by-Step Guide to Understanding [Subject]** — Step 1: [Initial focus or background]. Learn that [Subject Description].
Recipe	**Recipe for [General Concept]: [Specific Edition/Style]** — *Ingredients:* [Variable 1], [Variable 2], [Variable 3]...
Encyclopedia article	**[Subject]** is a [type/category] that [provides a description or function], [additional information if applicable].
Research article	This article explores the significance of [subject or topic], a [description or classification], characterized by [notable features or contributions].
Description of a thing or person	Introducing [Subject/Entity], a [descriptor] [type/category] [context/detail] renowned for its [property/characteristic].
FAQ	**What is [Subject]?** — [Subject] is a [general category or description] [specific type or detail] [additional information].
Legal terms & conditions	**Terms and Conditions Regarding [Subject/Theme]**.
Opinion	In my view, [Subject/Entity] represents [significance/impact/legacy] in [field/area], and its influence on [audience/community/context] cannot be overstated.
Review	[Subject] is a [descriptor] that [verb phrase] [contextual information].
Opinion blog (editorial)	When we think of [general category or field], [a notable example or subject] often comes to mind.
Denominational religious sermon	Beloved congregation, today we gather to reflect upon [individual/concept] that illuminates our lives and encourages us to contemplate our shared journey.
Description with intent to sell	Introducing [Subject]: a [descriptor] [product/service] designed for [use case]; discover how it [benefit/outcome] for [target user].
Informational persuasion	In the context of [domain or field], few [types/categories] resonate as profoundly within [subfields] as [specific work/name/entity].
Informational description	[Entity/Subject] is a [description] in the field of [broader category], specifically within [subcategory/locale].
News & opinion blog or editorial	When we think of [general category or field], [notable subject] often comes to mind — situating today's discussion of [topic] within [context].

training data) against "non-seen" splits (i.e., non-training data), which contain documents collected after the model's release and thus not used in its pretraining [9]. This construction makes the "non-seen" splits suitable as reliable non-training references in our evaluation. The MIMIR dataset is publicly available at https://huggingface.co/datasets/iamgroot42/mimir.

G Experimental Setup for Laundering Reversal Evaluation

G.1 Evaluation Metrics

Following Carlini et al. [8], we adopt three complementary metrics to evaluate membership inference attacks.

Area Under the ROC Curve (AUC). AUC measures the overall discriminative power of the attack, independent of any specific threshold. It reflects how well an unauthorized training data detection method can separate training data from unseen data on average. It may overstate effectiveness since it also includes high false-positive regions that are less relevant in practice.

Attack Success Rate (ASR). ASR measures the fraction of correctly identified training data under a single decision threshold that maximizes balanced accuracy across training data and unseen data. Unlike AUC, ASR reflects the practical effectiveness of an attack when deployed, as real-world unauthorized training data detectors typically operate at a single fixed threshold.

True Positive Rate at 5% False Positive Rate (TPR@5%). This metric evaluates the ability of a detector to identify training data while maintaining a strict false-positive constraint. Prior work highlights that low false-positive regimes are the most meaningful for privacy evaluation, since even a small number of incorrect training data decisions can undermine the credibility of the attack. TPR@5% therefore provides a high-precision view of attack success.

G.2 Experimental Setup

We evaluate the effectiveness of our approach by examining whether the SDR-synthesized data enhances the performance of existing unauthorized training data detection methods against LLMs trained on synthesized data. Specifically, we first sample 200 data points from the dataset that have not been seen by the *target* LLM. These 200 samples are synthesized into a new version using a prompt to simulate the data laundering process applied by the model provider. We then randomly select half of these synthesized samples as training data to fine-tune the LLM (training details are provided in Table 11), while the remaining half serves as non-training data. Subsequently, SDR is employed to infer a transformation prompt that restores the synthesis process and recovers the synthesized samples. Using this inferred prompt, we rewrite the original 200 data. Finally, we apply baseline unauthorized training data detectors to both the original 200 samples and the 200 samples rewritten with the inferred prompt, comparing their attack performance to assess whether the inferred prompt improves detection effectiveness.

Table 11: Hyperparameter settings for LoRA fine-tuning.

Lora_alpha	r	epochs	lr	gradient accumulation steps	weight_decay	logging_strategy
32	8	30	0.0004	25	0.1	"epoch"

Computational Resources. All experiments were conducted on Nvidia A100 GPUs with 80 GB of memory per GPU. We fine-tuned the target LLMs using LoRA, with the hyperparameters reported in Table 11. Each fine-tuning run used 1 GPU. The SDR pipeline was executed using 4 GPU(s) for target-model inference, while auxiliary LLMs were accessed via APIs for data synthesis and prompt refinement. The runtime and API query costs under different hyperparameter configurations are reported in Appendix M.

H Synthesized Prompts Used in Experiments

Table 12 lists the complete set of inside-register prompts corresponding to the 23 predefined registers in our framework. Each prompt is generated by GPT-5 using the instruction: "Generate a rewriting prompt that transfers the text into [REGISTER]." Here, [REGISTER] denotes one of the 23 registers. Table 13 lists the complete set of outside-register prompts; none align with any of the 23 predefined registers. These prompts are generated by GPT-5 using the instruction: "You are a prompt generator. Generate rewriting prompts that transform the original content into broadly different domains (e.g., Facebook post, academic article, children's story, legal document). Make the prompts as mutually distinct in domain as possible, and ensure each induces substantial changes to the original text rather than merely surface-level edits."

I The Specific Results for Each Prompt

Table 2 reports the average performance of each unauthorized training data detection method across models fine-tuned on data synthesized with different prompts. Tables 14, 15 and 16 present the detailed average performance across five unauthorized training data detection methods for each

Table 12: Inside register prompts.

ID	Laundering Prompts
1	Rewrite the text in a lyrical style, ensuring the imagery is vivid, the rhythm flows naturally.
2	Rewrite the text in a spoken style, making it sound natural and conversational, and ensure the tone feels engaging and easy to follow for a live audience.
3	Rewrite the text in the form of an interview, ensuring the questions flow naturally and the answers provide clear, engaging explanations for the audience.
4	Rewrite the text as an interactive discussion between two or more participants, ensuring the conversation flows logically, with each speaker's tone and style clearly distinguishable.
5	Rewrite the text as a storytelling narrative. The story should flow naturally, use simple and engaging language, and be easy for all kinds of listeners to follow.
6	Rewrite the text in the style of a news report, ensuring the information is presented objectively and concisely.
7	Rewrite the text as a sports report, ensuring the action is described with dynamic, energetic language that conveys the pace, tension, and excitement of the event.
8	Rewrite the text as a narrative blog post, organized into clear sections with subheadings. Use a tone that is engaging and reflective, blending storytelling with explanation.
9	Rewrite the text as a step-by-step instructional guide. Break the content into numbered steps, with each step beginning with a clear imperative verb.
10	Rewrite the text as a recipe, introduce the information as sequential steps.
11	Rewrite the text to persuade the reader through factual information, making sure to include at least three specific data points or statistics to support the argument.
12	Rewrite the text as a sales description, and be sure to include a clear call-to-action at the end.
13	Rewrite the text in the style of an editorial, making sure to include a clear stance or opinion and a concluding paragraph that calls for action or reflection.
14	Rewrite the text as an informational description, ensuring the tone is neutral and objective, and include at least one definition or clarification to help the reader better understand the subject.
15	Rewrite the text in the style of an encyclopedia entry, maintaining a neutral, authoritative tone, and include at least one date, fact, or reference to give it the appearance of being sourced.
16	Rewrite the text as an academic research article, structured with sections such as Abstract, Introduction, Method, Results, and Conclusion, and include at least one in-text citation (invented if necessary) to simulate scholarly referencing.
17	Rewrite the text as a descriptive profile of a specific thing or person, using vivid details and attributes (appearance, characteristics, or context) and ending with a short summary sentence that highlights its significance.
18	Rewrite the text in the form of a Frequently Asked Questions (FAQ) section, making sure to include at least three question–answer pairs, with the questions phrased from the perspective of a curious reader.
19	Rewrite the text as legal terms and conditions, using formal legal language, and ensure at least one numbered clause is included for clarity.
20	Rewrite the text as a personal opinion piece, written in the first person, making sure to clearly express a stance and support it with at least one reason or example.
21	Rewrite the text as a review, giving it a clear positive or negative stance, and include at least one specific detail or example to justify the evaluation.
22	Rewrite the text as an opinion blog post, written in a conversational and persuasive tone, and include at least one personal anecdote or illustrative example to strengthen the argument.
23	Rewrite the text as a denominational religious sermon, using a reverent and exhortative tone, and include at least one scriptural quotation or moral teaching to guide the audience toward reflection or action.

inside-synthesized prompt. Table 17 presents the details for each outside-synthesized prompt[4]. The "Original Prompt" column denotes the true synthesized prompt. The "Reversed Prompt" column denotes the best prompt recovered by SDR. The "Orig." column reports the average AUC across detection methods using the original data, whereas the "SDR" column reports the results using SDR-reversed data.

J Negative-control experiment

In the negative-control experiment, neither D_{pro} nor any of its laundered variants is included in the target model's training data. As shown in Table 18, SDR does not produce a meaningful improvement in detection performance, with AUC and ASR remaining near chance level.

[4]We acknowledge that SDR is not guaranteed to improve detection performance for every possible outside-register laundering prompt, particularly when the transformation differs substantially from the predefined register taxonomy.

Table 13: Outside register prompts.

ID	Laundering Prompts
1	Rewrite the following content as slide presentation bullet points. Focus on summarizing the key arguments and findings clearly and concisely. Use concise phrases that highlight core points.
2	Rewrite the following text in the style of a Facebook post. Sharing interesting information with followers. You may add light commentary, questions to the audience, or casual phrasing, but keep it natural and human-like. Avoid using emojis, hashtags, or overly dramatic expressions.
3	Adapt the text into a poetic form with vivid metaphors, rhythmic structure, and emotionally evocative language.
4	Convert the content into a tutorial-style explanation for beginners, using step-by-step instructions, simple analogies, and common misunderstandings.
5	Rewrite the text as a formal business email, ensuring clarity, professionalism, and a polite tone.
6	Rewrite the passage as a scientific abstract, including Background, Methods, Results, and Conclusions. Invent at least two numerical values (percentages, sample sizes, or statistical outcomes) to support claims.
7	Rewrite the text as a product description for an e-commerce website, highlighting key features, benefits, and use cases in a persuasive manner.
8	Rewrite the text as a blog post, incorporating vivid descriptions of locations, cultural insights, and personal experiences to engage readers.
9	Rewrite the text as a classroom lecture transcript, with explanations, rhetorical questions, and occasional student interaction.
10	Convert the content into a job interview answer where the speaker uses it to explain their experience or perspective.

K SDR Transferability to Third-Party Laundering Pipelines

Alternative LLM Launderer with DeepSeek. We fine-tune the target LLM exclusively on data laundered by DeepSeek-v3 [20], while SDR continues to use GPT-4o-mini as the auxiliary model M_a for prompt search. We apply the first five inside-register prompts (Table 12) as laundering templates and execute laundering with DeepSeek-v3. As shown in Table 19, SDR consistently improves detection performance across all four off-the-shelf detectors. For example, Loss improves from AUC/ASR of $0.65/0.64$ to $0.81/0.76$, and Min-K improves from $0.67/0.65$ to $0.78/0.73$. This demonstrates that SDR can recover useful transformations even when the laundering pipeline is completely different from GPT-4o-mini.

Human as Launderer (Polite Dataset). While LLMs enable automated, large-scale rewriting practices, real-world laundering may not be carried out solely by LLMs. We simulate a laundering scenario using the Polite dataset [31], in which human annotators rewrote impolite sentences into polite versions. In this experiment, we treat the polite versions as "laundered" training data D_{train}, while their original versions are treated as D_{pro}. We find that SDR recovers a “personal review" style prompt (as shown below) and improves most off-the-shelf detectors (from Table 20).

We notice that Recall+SDR becomes slightly worse than Recall alone, which we attribute to polite expressions such as “I think” being already common in pretraining, making both training and non-training data equally easy to continue and collapsing the gap Recall relies on. Other detectors (+SDR) remain generally effective. In practice, Loss (+SDR) and Min-K (+SDR) may provide more reliable signals in this setting. Overall, these results demonstrate that SDR remains effective when the laundering pipeline uses a different LLM or human rewriting, and when the auxiliary and laundering models are mismatched, confirming the transferability of SDR in diverse and practical scenarios.

L Mixed-Register Laundering Pipelines

We have conducted experiments on **mixed-register transformations** by combining two registers in the laundering prompt (e.g., “opinion blog post with a persuasive tone”, “storytelling narrative as a sports report”, “informational description like a recipe”). Across all three mixed prompts, SDR consistently improves detection. For each mixed prompt, we present the recovered prompt followed by the corresponding detection results. **Mixed-register prompt 1.** Rewrite the text as an **opinion blog post** with a **persuasive tone**. *SDR-reversed prompt:* Rewrite the text as an opinion blog post, emphasizing personal narratives and reflections that highlight the emotional, cultural, or historical significance of the subject while engaging the reader. **Mixed-register prompt 2.** Rewrite the text as a **storytelling narrative** to introduce information as a **sports report**. *SDR-reversed prompt:* Rewrite the text as a narrative blog post. Focus on transforming the original information into a more engaging story, emphasizing the historical context and the evolution of the organization while

Table 14: Inside register prompts and corresponding reversed prompts with Orig and SDR results.

ID	Original Prompt	Reversed Prompt (SDR)	Orig (AUC)	SDR (AUC)
1	Rewrite the text in a lyrical style, ensuring the imagery is vivid, the rhythm flows naturally.	Rewrite the text in a lyrical style, transforming factual statements into evocative poetic expressions that utilize imagery and rhythm to capture the essence and emotional depth of the subject.	0.545	0.770
2	Rewrite the text in a spoken style, making it sound natural and conversational, and ensure the tone feels engaging and easy to follow for a live audience.	Rewrite the text in a natural and conversational spoken style, simplifying complex ideas and language while incorporating personal touches and engaging elements to enhance relatability and listener engagement.	0.716	0.916
3	Rewrite the text in the form of an interview, ensuring the questions flow naturally and the answers provide clear, engaging explanations for the audience.	Rewrite the text in the form of an interview, transforming the original information into a conversational dialogue that utilizes a question-and-answer structure to highlight key themes, significance, and insights.	0.726	0.876
4	Rewrite the text as an interactive discussion between two or more participants, ensuring the conversation flows logically, with each speaker's tone and style clearly distinguishable.	Rewrite the text as an interactive discussion between two or more participants, transforming the original content into a conversational format that incorporates questions, responses, and engagement while exploring the key themes and details.	0.647	0.798
5	Rewrite the text as a storytelling narrative. The story should flow naturally, use simple and engaging language, and be easy for all kinds of listeners to follow.	Rewrite the text as a vivid and engaging storytelling narrative that transforms factual information into an immersive experience, incorporating emotional depth, character-driven elements, and descriptive language to bring the subject to life.	0.633	0.754
6	Rewrite the text in the style of a news report, ensuring the information is presented objectively and concisely.	Rewrite the text in the style of an encyclopedia entry, focusing on providing a clear, concise, and structured overview of the subject while maintaining a formal tone and emphasizing relevant details and historical significance.	0.777	0.897
7	Rewrite the text as a sports report, ensuring the action is described with dynamic, energetic language that conveys the pace, tension, and excitement of the event.	Rewrite the text as a dynamic sports report, emphasizing key achievements, competitive aspects, and engaging narratives with vivid language and an exciting tone to capture the essence of the subject.	0.639	0.811
8	Rewrite the text as a narrative blog post, organized into clear sections with subheadings. Use a tone that is engaging and reflective, blending storytelling with explanation.	Rewrite the text as a Frequently Asked Questions section, transforming the original content into clear, concise questions and answers that highlight key information and maintain an engaging tone.	0.677	0.779
9	Rewrite the text as a step-by-step instructional guide. Break the content into numbered steps, with each step beginning with a clear imperative verb.	Rewrite the text as a step-by-step instructional guide, breaking down the information into clear, numbered steps that provide concise and informative insights on the specified topic.	0.783	0.895
10	Rewrite the text as a recipe, introduce the information as sequential steps.	Transform the original text into an interview format, utilizing a question-and-answer structure to convey the key information while maintaining clarity, engagement, and the essence of the subject's background or significance.	0.703	0.780
11	Rewrite the text to persuade the reader through factual information, making sure to include at least three specific data points or statistics to support the argument.	Rewrite the text as a review, focusing on summarizing key elements, highlighting significant contributions or achievements, and emphasizing the broader impact or emotional resonance while maintaining an engaging narrative style.	0.632	0.751

Table 15: Inside register prompts and corresponding reversed prompts with Orig and SDR results (continues).

ID	Original Prompt	Reversed Prompt (SDR)	Orig (AUC)	SDR (AUC)
12	Rewrite the text as a sales description, and be sure to include a clear call-to-action at the end.	Rewrite the text as a sales description, creating an engaging narrative that emphasizes the subject's achievements, significance, and unique qualities, while encouraging readers to explore further and take action.	0.622	0.817
13	Rewrite the text in the style of an editorial, making sure to include a clear stance or opinion and a concluding paragraph that calls for action or reflection.	Rewrite the text as an opinion blog post that transforms factual content into a compelling narrative, emphasizing historical or cultural significance, while encouraging reader engagement and reflection on relevant issues.	0.582	0.703
14	Rewrite the text as an informational description, ensuring the tone is neutral and objective, and include at least one definition or clarification to help the reader better understand the subject.	Rewrite the text in the style of an encyclopedia entry, focusing on providing a clear, concise, and structured overview of the subject, emphasizing key details, historical context, and maintaining a formal and neutral tone.	0.791	0.929
15	Rewrite the text in the style of an encyclopedia entry, maintaining a neutral, authoritative tone, and include at least one date, fact, or reference to give it the appearance of being sourced.	Rewrite the text in the style of an encyclopedia entry, focusing on enhancing clarity, structure, and formal tone while providing relevant historical context and highlighting key contributions or significance.	0.795	0.942
16	Rewrite the text as an academic research article, structured with sections such as Abstract, Introduction, Method, Results, and Conclusion, and include at least one in-text citation (invented if necessary) to simulate scholarly referencing.	Rewrite the text as a recipe.	0.475	0.493
17	Rewrite the text as a descriptive profile of a specific thing or person, using vivid details and attributes (appearance, characteristics, or context) and ending with a short summary sentence that highlights its significance.	Rewrite the text to persuade the reader through factual information, enhancing the narrative with vivid descriptions and emphasizing the significance and contributions of the subject while maintaining an engaging flow.	0.661	0.775
18	Rewrite the text in the form of a Frequently Asked Questions (FAQ) section, making sure to include at least three question–answer pairs, with the questions phrased from the perspective of a curious reader.	Rewrite the text in the style of a structured news report, emphasizing clarity, conciseness, and the presentation of key details through organized sections and formal tone.	0.704	0.781
19	Rewrite the text as legal terms and conditions, using formal legal language, and ensure at least one numbered clause is included for clarity.	Rewrite the text as legal terms and conditions, maintaining a formal tone and structure while incorporating legal terminology and clearly defined sections that organize the content into concise and coherent clauses.	0.744	0.890
20	Rewrite the text as a personal opinion piece, written in the first person, making sure to clearly express a stance and support it with at least one reason or example.	Rewrite the text as a step-by-step instructional guide.	0.564	0.546
21	Rewrite the text as a review, giving it a clear positive or negative stance, and include at least one specific detail or example to justify the evaluation.	Rewrite the text as a review, focusing on key achievements, contributions, or themes while maintaining a positive and engaging tone that emphasizes impact and personal evaluation.	0.583	0.715

Table 16: Inside register prompts and corresponding reversed prompts with Orig and SDR results (continues).

ID	Original Prompt	Reversed Prompt (SDR)	Orig (AUC)	SDR (AUC)
22	Rewrite the text as an opinion blog post, written in a conversational and persuasive tone, and include at least one personal anecdote or illustrative example to strengthen the argument.	Rewrite the text in the form of a Frequently Asked Questions section, emphasizing key details and themes while maintaining a clear, engaging, and conversational tone.	0.592	0.667
23	Rewrite the text as a denominational religious sermon, using a reverent and exhortative tone, and include at least one scriptural quotation or moral teaching to guide the audience toward reflection or action.	Rewrite the text as a step-by-step instructional guide that transforms the content into clear, actionable insights while maintaining an engaging and motivational tone that encourages personal reflection and connection.	0.532	0.566

maintaining a conversational tone. **Mixed-register prompt 3.** Rewrite the text as an **informational description** like a **recipe**. *SDR-reversed prompt:* Rewrite the text as a step-by-step instructional guide, clearly outlining key aspects and maintaining a logical flow and numbered steps.

Across all three mixed-register prompts, SDR consistently improves both AUC and ASR, demonstrating that it can successfully transfer to mixed laundering transformations. Nevertheless, we agree that extremely exotic transformations (e.g., pseudo-translation into low-resource languages) may further stress SDR.

M Hyperparameter Sensitivity Study

We conducted an extended sensitivity study to examine how SDR behaves under different choices of the key hyper-parameters K (see Table 22), l (see Table 23), m (see Table 24), and n (see Table 25), using Loss+SDR under the No. 1 inside-register prompt.

Our key findings are as follows (see the tables below for detailed numbers): Increasing K from $3 \rightarrow 15$ improves AUC from $0.71 \rightarrow 0.75$ and ASR from $0.69 \rightarrow 0.71$, at the cost of higher runtime and query budget (roughly \$7–\$39 with GPT-4o; substantially cheaper with GPT-4o-mini). The value of m has the strongest impact. Varying m from $3 \rightarrow 9$ raises AUC from $0.72 \rightarrow 0.81$ and ASR from $0.69 \rightarrow 0.76$ with almost unchanged query cost, making $m = 9$ a strong default choice. l and n exhibit moderate gains. Setting $l = 7$ and $n = 7$ achieves a good balance between performance and cost. *Note that n is used only once when building templates and does not affect per-run latency.*

Detailed numerical results. Unless otherwise specified, we use $K = 10$, $l = 5$, $m = 5$, and $n = 10$ as the default hyperparameters throughout our experiments. Based on the sensitivity analysis, we recommend $(K = 5, l = 7, m = 9, n = 7)$ as a potentially more efficient configuration that balances detection performance and computational cost.

N Comparison with Direct Optimization

To further evaluate the effectiveness of the structured two-stage search in SDR, we compare it with a baseline that directly optimizes Equation 1 under the same budget of detection-performance evaluations.

Direct Search. The baseline directly searches over the synthesis prompt space to maximize detection performance. Starting from a randomly initialized prompt generated by GPT-4o-mini, we synthesize candidate texts and evaluate their detection performance on the target model. At each subsequent iteration, we provide GPT-4o-mini with all previously evaluated prompts and their corresponding performance scores, asking it to propose a new prompt that may further improve detection performance. The best-performing prompt is selected after exhausting the evaluation budget.

Budget Matching. Each SDR reversion pipeline requires 15 full detection-performance evaluations: five evaluations for shortlisted registers in Algorithm 1 and ten additional evaluations for newly

Table 17: Outside register prompts and corresponding reversed prompts with Orig and SDR results.

ID	Original Prompt	Reversed Prompt (SDR)	Orig	SDR
1	Rewrite the following content as slide presentation bullet points. Focus on summarizing the key arguments and findings clearly and concisely. Use concise phrases that highlight core points.	Rewrite the text as a step-by-step instructional guide, breaking down the information into clear and organized steps that highlight key aspects, milestones, and contributions relevant to the subject matter.	0.722	0.796
2	Rewrite the following text in the style of a Facebook post. Sharing interesting information with followers. You may add light commentary, questions to the audience, or casual phrasing, but keep it natural and human-like. Avoid using emojis, hashtags, or overly dramatic expressions.	Rewrite the text in the form of an interview, transforming it into a conversational format that incorporates engaging questions and answers while maintaining the essence of the original content.	0.618	0.696
3	Adapt the text into a poetic form with vivid metaphors, rhythmic structure, and emotionally evocative language.	Rewrite the text in a lyrical, poetic style, transforming factual or narrative content into evocative poetic expression through vivid imagery, metaphor, rhythm, and emotional depth while preserving the essence, meaning, and key details of the original content.	0.527	0.673
4	Convert the content into a tutorial-style explanation for beginners, using step-by-step instructions, simple analogies, and common misunderstandings.	Rewrite the text as an interactive discussion between two or more participants, transforming factual information into a conversational format that encourages questions, responses, and engagement while highlighting key points and significance.	0.610	0.673
5	Rewrite the text as a formal business email, ensuring clarity, professionalism, and a polite tone.	Rewrite the text as a Frequently Asked Questions section, structuring it into clear questions and answers that summarize key information and enhance understanding of the subject matter.	0.804	0.901
6	Rewrite the passage as a scientific abstract, including Background, Methods, Results, and Conclusions. Invent at least two numerical values (percentages, sample sizes, or statistical outcomes) to support claims.	Rewrite the text as a descriptive profile of a specific thing or person, focusing on enhancing the narrative structure and providing a comprehensive overview of their background, achievements, and contributions in a clear and engaging manner.	0.597	0.648
7	Rewrite the text as a product description for an e-commerce website, highlighting key features, benefits, and use cases in a persuasive manner.	Rewrite the text as a Frequently Asked Questions section, transforming the original information into a clear and engaging question-and-answer format that highlights key features, significance, and provides concise, informative responses.	0.604	0.705
8	Rewrite the text as a blog post, incorporating vivid descriptions of locations, cultural insights, and personal experiences to engage readers.	Rewrite the text into a Frequently Asked Questions section, transforming the original content into a clear and engaging question-and-answer format while highlighting key details, milestones, and contextual information.	0.625	0.704
9	Rewrite the text as a classroom lecture transcript, with explanations, rhetorical questions, and occasional student interaction.	Rewrite the text as a Frequently Asked Questions section, transforming the information into clear, concise questions and answers that highlight key aspects relevant to the subject matter.	0.674	0.775
10	Convert the content into a job interview answer where the speaker uses it to explain their experience or perspective.	Rewrite the text in the style of an editorial, enhancing the narrative by emphasizing significant themes, broader context, and community aspects, while maintaining an engaging and reflective tone throughout.	0.574	0.680

Table 18: Negative-control experiment on **Pythia-6.9B**. When the target model is *not* trained on any laundered variants of the proprietary data, SDR does not find a prompt to improve unauthorized training-data detection. Across all detectors and both datasets (ArXiv and Wikipedia), the AUC and ASR scores before and after SDR remain close to 0.5, indicating that SDR does not fabricate evidence of misuse in the absence of laundered training data.

Method	ArXiv		Wikipedia	
	AUC	ASR	AUC	ASR
Recall	0.516	0.532	0.528	0.545
Recall + SDR	0.488	0.525	0.473	0.540
Loss	0.486	0.522	0.471	0.510
Loss + SDR	0.490	0.535	0.465	0.515
Min-K	0.536	0.510	0.486	0.515
Min-K + SDR	0.499	0.535	0.440	0.525
Min-K++	0.476	0.521	0.483	0.535
Min-K++ + SDR	0.483	0.520	0.446	0.515

Table 19: SDR transferability on DeepSeek-v3 laundering pipelines for the target LLM trained exclusively on DeepSeek-laundered data. SDR uses GPT-4o-mini as M_a.

Method	AUC	ASR
Recall	0.641	0.630
Recall + SDR	**0.767**	**0.718**
Loss	0.650	0.635
Loss + SDR	**0.808**	**0.760**
Min-K	0.672	0.645
Min-K + SDR	**0.784**	**0.734**
Min-K++	0.555	0.584
Min-K++ + SDR	**0.565**	**0.595**

proposed prompts in Algorithm 2, with the current prompt score cached across iterations. For a fair comparison, we allocate the direct search baseline the same budget of 15 evaluations.

Experimental Setup. We conduct experiments on the first three inside-register laundering prompts using MIMIR-Wikipedia and Pythia-6.9B as the target model. GPT-4o-mini serves as the auxiliary model for prompt generation and data rewriting. We evaluate three detection methods: Recall, Loss, and Min-K, using AUC and ASR as evaluation metrics.

Results and Analysis. As shown in Table 26, direct optimization of Equation 1 provides limited or even negative improvements over the original detectors under the same computational budget. In contrast, SDR consistently achieves the highest AUC and ASR across all three laundering prompts and all detection methods. These results demonstrate that directly searching the unconstrained prompt space is inefficient under a limited evaluation budget, whereas SDR's goal-detail decomposition enables more effective exploration of detection-useful synthesis transformations.

O Comparison to Reverse Prompt Engineering Methods

To clarify how SDR differs from related lines of work, especially those based on prompt-recovery techniques, we note an important methodological distinction: To the best of our knowledge, existing *reverse prompt engineering* methods assume the unknown prompt is applied at *inference* time and leaves a direct trace in the observed outputs. In our setting, the laundering prompt is applied before *training*. The data-rights holder only sees the trained model and never observes any text generated under the unknown laundering prompt, making a direct application impossible.

We summarize the conceptual differences between Reverse Prompt Engineering [15] and SDR as follows. Reverse prompt engineering assumes the unknown prompt is applied during inference,

Table 20: Detection performance on the Polite (human-rewritten) dataset. SDR improves most detectors even when laundering is performed by humans, demonstrating SDR's robustness and transferability beyond LLM-based pipelines.

Method	AUC	ASR
Recall	**0.649**	**0.635**
Recall + SDR	0.566	0.575
Loss	0.687	0.680
Loss + SDR	**0.751**	**0.725**
Min-K	0.652	0.680
Min-K + SDR	**0.713**	**0.690**
Min-K++	0.560	0.575
Min-K++ + SDR	**0.624**	**0.615**

Table 21: SDR transferability on mixed-register laundering prompts. SDR consistently improves AUC and ASR across three mixed-register laundering transformations, demonstrating robustness to more complex combinations of stylistic rewrites.

Method	Mixed Prompt 1		Mixed Prompt 2		Mixed Prompt 3	
	AUC	ASR	AUC	ASR	AUC	ASR
Recall	0.559	0.560	0.666	0.650	0.737	0.715
Recall + SDR	**0.757**	**0.705**	**0.777**	**0.725**	**0.831**	**0.770**
Loss	0.572	0.575	0.670	0.655	0.738	0.685
Loss + SDR	**0.767**	**0.705**	**0.796**	**0.750**	**0.834**	**0.785**
Min-K	0.587	0.580	0.677	0.670	0.747	0.705
Min-K + SDR	**0.750**	**0.715**	**0.749**	**0.700**	**0.820**	**0.765**
Min-K++	0.475	0.535	0.475	0.515	0.586	0.605
Min-K++ + SDR	**0.586**	**0.615**	**0.538**	**0.580**	**0.617**	**0.615**

meaning it directly conditions the model's outputs and leaves observable traces that the recovery procedure can exploit. In contrast, SDR targets laundering transformations applied before training, where the prompt does not manifest in any generated text. Instead, only implicit consequences of the laundering process remain embedded in the trained model. Moreover, reverse prompt engineering methods rely on having direct access to model outputs produced under the unknown prompt, whereas in our setting, the data-rights holder never observes any prompt-conditioned outputs and can only query the final trained model. These differences make existing reverse prompt engineering approaches incompatible with the laundering-before-training scenario that SDR is designed to address.

To still provide a comparison, we slightly adapt Reverse Prompt Engineering [15] to a use case better aligned with our problem setup: given an original document, we let M_t generate a continuation and treat that continuation as a proxy "laundered" version; we then ask Reverse Prompt Engineering [15] to infer a prompt connecting the original and continuation.

Table 27 shows that SDR provides the largest improvement among all methods. Loss+SDRachieves an AUC of 0.782 and an ASR of 0.764, clearly outperforming both the baseline detector and the adapted reverse prompt engineering approach. While reverse prompt engineering offers modest gains over the baseline, its improvements remain limited, highlighting that it is considerably less effective than SDR in recovering laundering transformations and restoring detector performance.

P TPR@1% Result

We report TPR@1% (True Positive Rate at 1% False Positive Rate) as a complementary metric to TPR@5% reported in the main paper. While TPR@5% evaluates detection performance under a relatively lenient false-positive constraint, TPR@1% imposes a stricter threshold, reflecting scenarios where auditors require higher precision before making an accusation of data misuse. A higher TPR@1% indicates that the detection method can correctly identify more training samples while maintaining a very low rate of false accusations. As shown in Table 28, SDR consistently improves

Table 22: Sensitivity to K.

K	AUC	ASR	Time (hh:mm:ss)	Query Budget (GPT4o / GPT4o-mini)
3	0.712	0.685	02:23:42	$\sim$ \$7/0.8
5	0.727	0.690	04:11:33	$\sim$ \$13/1.5
15	0.747	0.705	13:27:12	$\sim$ \$39/4.5

Table 23: Sensitivity to l.

l	AUC	ASR	Time (hh:mm:ss)	Query Budget (GPT4o / GPT4o-mini)
3	0.715	0.670	08:21:33	$\sim$ \$10/1
7	0.731	0.695	09:13:12	$\sim$ \$10/1
9	0.718	0.695	09:35:17	$\sim$ \$10/1

TPR@1% across all detectors and both inside- and outside-register settings, further confirming its effectiveness under stricter evaluation criteria.

Q Partial Laundering Experiment

We conducted an experiment to evaluate whether SDR remains effective when only a portion of the proprietary data is laundered. Given $|D_{\text{pro}}| = 200$, we randomly selected half of the samples and applied the first five inside-register prompts (Appendix Table 9) to generate the laundered subsets used for training, while the remaining samples were left unaltered. SDR was then applied to the full D_{pro} to infer the laundering transformation. Table 29 reports the average performance across all five prompts. SDR successfully improves detection performance across all four off-the-shelf detectors. For example, the AUC/ASR of Loss increases from 0.599/0.597 to 0.732/0.703.

R Ethics Statement

This work focuses on defending against unauthorized data laundering in LLM training by restoring the effectiveness of post-hoc detection. All experiments were conducted on publicly available datasets, and no proprietary or personal data was used. While there is a risk that our techniques could be misused to improve laundering attacks, our contributions are explicitly framed for defensive purposes, aiming to strengthen accountability and responsible governance in AI systems.

S Reproducibility Statement

We have made significant efforts to ensure the reproducibility of our results. All datasets used in this paper are publicly available, and details of data synthesis procedures are provided in Appendix H. Complete descriptions of model architectures, training settings, and evaluation protocols are included in the main paper and Appendix G. For each experiment, we specify hyper-parameters, implementation details, and the auxiliary LLM prompts used for synthesis and reverse synthesis. Our codebase, built on PyTorch and Hugging Face Transformers, was included in the submitted supplementary material. Our implementation is publicly available at https://github.com/MuxingLiUnimelb/SDR, together with instructions for reproducing the main experiments.

T AI usage Clarification

Large Language Models improved the manuscript's grammar and readability; all research design, analysis, and interpretation were conducted by the authors.

Table 24: Sensitivity to m.

m	AUC	ASR	Time (hh:mm:ss)	Query Budget (GPT4o / GPT4o-mini)
3	0.718	0.685	08:14:33	$\sim$ \$10/1
7	0.796	0.740	08:19:29	$\sim$ \$10/1
9	0.810	0.760	08:26:45	$\sim$ \$10/1
11	0.806	0.755	08:33:34	$\sim$ \$10/1

Table 25: Sensitivity to n. n controls the construction of the template and is computed only once during initialization. Therefore, it does not contribute to the runtime or query budget.

n	AUC	ASR	Time (hh:mm:ss)	Query Budget
3	0.705	0.675	–	–
7	0.752	0.705	–	–
9	0.742	0.695	–	–

Table 26: Comparison of direct Equation 1 optimization and SDR under the same computational budget on three inside-register laundering prompts. The best results are highlighted in bold.

	Prompt 1		Prompt 2		Prompt 3	
Method	AUC	ASR	AUC	ASR	AUC	ASR
Recall	0.538	0.550	0.690	0.680	0.697	0.675
Recall + Direct Equation 1	0.556	0.565	0.689	0.655	0.567	0.590
Recall + SDR	**0.698**	**0.680**	**0.895**	**0.810**	**0.856**	**0.795**
Loss	0.515	0.545	0.697	0.675	0.707	0.685
Loss + Direct Equation 1	0.579	0.585	0.695	0.675	0.593	0.595
Loss + SDR	**0.727**	**0.690**	**0.921**	**0.870**	**0.856**	**0.800**
Min-K	0.551	0.580	0.696	0.670	0.708	0.690
Min-K + Direct Equation 1	0.553	0.590	0.663	0.640	0.608	0.610
Min-K + SDR	**0.746**	**0.715**	**0.899**	**0.835**	**0.857**	**0.775**

Table 27: Comparison between SDR and reverse prompt engineering.

Method	AUC	ASR
Loss + SDR	**0.782**	**0.764**
Loss + Reverse Prompt Engineering	0.682	0.663
Loss	0.638	0.635

Table 28: TPR@1% results for the main experiment.

Method	Inside TPR@1%	Outside TPR@1%
Recall	0.001	0.000
Recall + SDR	**0.174**	**0.038**
Loss	0.001	0.000
Loss + SDR	**0.164**	**0.054**
Ref	0.027	0.021
Ref + SDR	**0.137**	**0.035**
Zlib	0.001	0.001
Zlib + SDR	**0.072**	**0.050**
Min-K	0.001	0.001
Min-K + SDR	**0.151**	**0.064**

Table 29: Average performance across all five laundering prompts under partial laundering.

Method	Avg AUC	Avg ASR
Recall	0.608	0.590
Recall + SDR	**0.722**	**0.694**
Loss	0.599	0.597
Loss + SDR	**0.732**	**0.703**
Min-K	0.608	0.603
Min-K + SDR	**0.724**	**0.701**
Min-K++	0.553	0.565
Min-K++ + SDR	**0.580**	**0.602**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see the last paragraph of the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: please see Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not include theoretical results or proofs. All formulas used in the paper are adopted from prior work and properly cited.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, all experimental settings required to reproduce our results, including training hyperparameters and prompts used, are fully described in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our implementation is publicly available at https://github.com/MuxingLiUnimelb/SDR, together with instructions for reproducing the main experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes, all training and test details, including hyperparameters and optimizer settings, are fully specified in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report average detection performance across multiple laundering prompts to evaluate the robustness of SDR under diverse transformation settings. However, we do not provide formal statistical significance tests or confidence intervals. We acknowledge this limitation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the compute resources used in our experiments, including hardware specifications and execution time, are detailed in Appendix M.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes, the research conducted in this paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, this paper discusses broader impacts. Our work has a positive societal impact by protecting data privacy and supporting data rights holders against unauthorized use of their data for LLM training. Potential negative impacts, such as adversarial misuse of the proposed method, are also discussed in the paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release any models or datasets that pose a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all datasets and models used in this paper are properly cited with references to the original papers. The licenses and terms of use for all assets are explicitly mentioned and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce or release any new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects, and therefore does not require IRB approval.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Yes, LLMs are used as a core component of our method. Specifically, we employ LLMs as auxiliary rewriting models to perform detection.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.